

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings of claims in the application:

LISTING OF CLAIMS:

1. (original) A method for carrying out a connection migration in a multi-computer architecture (cluster), from a first node, called primary node, comprising a first computer in said cluster whereon an initial software application is executed, towards at least one secondary node, comprising another computer in said cluster, characterised in that it implements a virtual network address which is carried by the first computer and which is transferred to the other computer, said virtual network address being provided as a dialogue link between the cluster and clients computers connected to said cluster and affected by the software application.

2. (original) A migration method according to claim 1, wherein the connections are associated with a software application intended to be replicated on at least one other computer in order to enable a service switch from the initial application to the replica thereof.

3. (currently amended) A migration method according to ~~one of the claims 1 or 2~~ claim 1, characterised in that the messages issued from a client are captured before being taken into account by the network layer of the cluster.

4. (original) A migration method according to claim 3, implemented in the context of a protocol TCP/IP, characterised in that the messages are captured at the "IP" tables level.

5. (currently amended) A migration method according to ~~any of the previous claims~~ claim 1, characterised in that the messages received on the virtual network address are transmitted to the secondary computer(s) over a multicast-type channel.

6. (currently amended) A migration method according to ~~the claims 4 and 5~~ claim 4, characterised in that it comprises capturing and modifying the socket parameters, via extended generic system calls.

7. (original) A migration method according to claim 6, characterised in that the captured and modified socket parameters include at least one of the following parameters:

- local and remote ports
- local and remote reference number
- number of the next packet to be emitted, and expected
- emission and reception timer
- window size

8. (original) A migration method according to claim 7, characterised in that it includes backing up the list of packages pending transmission (send queue), the packages in transit and the packages received, but which have not been read as yet by the application.

9. (original) A method for carrying out functional continuity of a software application in a multi-computer architecture (cluster), whereas said application is run at a given time on one of the computers of the cluster, called main node, whereas the other computers of said cluster are called secondary nodes, said method implementing the connection migration method according to any of the previous claims, characterised in that it comprises the following steps:

- on-the-flow updating of at least one clone of the application on at least one of the secondary nodes,
- when detecting a fault or an event affecting said main node, switching the service towards one at least of said clones, and
- migrating network connections.

10. (original) A functional continuity method according to claim 9, characterised in that it includes moreover updating the clones of the application.

11. (original) A functional continuity method according to claim 10, characterised in that the clones are updated periodically.

12. (currently amended) A functional continuity method according to ~~any of the claims 10 or 11~~ claim 10, characterised in that the updating of the clones of the application is triggered on one or several characteristic events.

13. (currently amended) A functional continuity method according to ~~any of the claims 9 to 12~~ claim 9, characterised in that it includes moreover supervision of the state of resources necessary to the operation of the application.

14. (currently amended) A functional continuity method according to ~~any of the claims 9 to 13~~ claim 9, characterised in that it includes moreover, when detecting a fault or an event affecting the main node, a step for electing, among clones installed on secondary nodes, a clone for being substituted for the initial application, whereas the node whereon said clone is installed becomes the new main node.

15. (currently amended) A functional continuity method according to ~~any of the claims 9 to 14~~ claim 9, characterised in that it includes moreover on each clone a record of the messages received by the primary node, whereas these messages will be re-injected into the clone elected as new primary if switching.

16. (currently amended) A multi-computer system provided to run on at least one of said computers at least one software application, implementing the method for carrying out functional continuity according to ~~any of the claims 9 to 15~~ claim 9.

17. (currently amended) An application of the connection migration method according to ~~any of the claims 1 to 8~~ claim 1, for automatic optimisation of data-processing resources by load sharing by dynamic method distribution.

18. (currently amended) An application of the connection migration method according to ~~any of the claims 1 to 8~~ claim 1, for non-interruptive maintenance by process re-location upon request, over an data-processing resource network.

19. (currently amended) An application of the connection migration method according to ~~any of the claims 1 to 8~~ claim 1, for preservation of applicative context in mobile applications.

20. (new) A migration method according to claim 5, characterised in that it comprises capturing and modifying the socket parameters, via extended generic system calls.